ARL-SR-0344 • Oct 2015

US Army Research Laboratory

Documentation of the Tonge–Ramesh Material Model for Release 2015-06-05-152756

by Andrew L Tonge

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-SR-0344 ● Oct 2015

US Army Research Laboratory

Documentation of the Tonge–Ramesh Material Model for Release 2015-06-05-152756

by Andrew L Tonge
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
October 2015	Final	October 2014–November 2014

4. TITLE AND SUBTITLE
Documentation of the Tonge–Ramesh
Material Model for Release 2015-06-05-152756

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Andrew L Tonge

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-WMP-C
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-SR-0344

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
primary author's email: <andrew.l.tonge.civ@mail.mil>.

14. ABSTRACT
This document serves as a user manual for the Tonge–Ramesh material model, implemented in EPIC, Alegra, and ALE3D. Each code can include portions of this manual as appropriate for the code-specific user manuals. It is updated from the user manual that appeared in Appendix B of the author's doctoral dissertation published by The Johns Hopkins University, to include input parameter changes since extracting the model from Uintah, the addition of a single surface granular flow model, and notes regarding the repackaging of the model as an Abaqus Umat for greater portability.

15. SUBJECT TERMS
ceramic, damage, material model, failure

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Andrew L Tonge
Unclassified	Unclassified	Unclassified	UU	34	19b. TELEPHONE NUMBER (Include area code) 410-278-1069

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

Acknowledgments

The material model was initially developed at The Johns Hopkins University with partial support from the Materials in Extreme Dynamic Environments (MEDE) program under the guidance of KT Ramesh. The material model was extracted from the original Uintah implementation as a part of a Productivity Enhancement and Technology Transfer (PETT) support project by Ross Smith. The suggestions and insight of Rich Becker were important for developing a portable material model.

INTENTIONALLY LEFT BLANK.

1. Introduction

The Tonge-Ramesh material model is a micromechanics based constitutive model for the high rate failure of quasi-brittle materials such as armor ceramics. It incorporates micromechanics based damage, granular flow of the damaged material, lattice plasticity (volume preserving), and equation of state coupling. The key physical aspects of the model are described in (Tonge 2014, ch. 4). This document serves to detail the user input parameters and additional functionality that is not discussed in the paper. This document is not intended to discuss the physical implications of the input parameters or discuss the physical reasoning used to develop the material model.

2. Material Model Inputs

The following list describes the model inputs. All inputs must be provided, and the current model does not do any error checking to ensure that reasonable values are provided for the user inputs. Units are described using the following abbreviations:

- P: Stress—Unless otherwise noted, scalar measures of stress and strain used in this model are the magnitude of the deviatoric part of the tensor and the magnitude of the isotropic part of the tensor. For a generic tensor $\boldsymbol{A}$, $r_A = \sqrt{\boldsymbol{A}_{\text{dev}} : \boldsymbol{A}_{\text{dev}}}$ and $z_A = \text{tr}(\boldsymbol{A})/\sqrt{3}$. The same measures of deviatoric and isotropic magnitude are used for both stress and strain. Constants from other models that use traditional measures like the Von–Mises equivalent stress and the pressure will need to be rescaled.
- L: Length
- τ: Time
- T: Temperature
- M: Mass
- E: Energy

1) `useDamage`: Unitless—Flag to activate damage calculation module; 0—do not perform damage calculation; 1—do damage calculation.

2) `usePlasticity`: Unitless—Flag to activate J_2 plasticity; 0—do not perform calculation; 1—perform calculation.

3) `useGranularPlasticity`: Unitless—Flag to activate pressure dependent granular flow calculations; 0—do not perform calculation; 1—perform calculation.

4) `useOldStress`: Unitless—Flag that controls sublooping in damage growth calculation; 1—do not do sublooping, instead use the stress from the beginning of the time step.

5) `artificialViscosity`: Unitless—Flag to activate linear and quadratic bulk viscosity calculation; 0—do not perform calculation; 1—perform calculation.

6) `artificialViscousHeating`: Unitless—Flag to convert work done by artificial viscous pressure to temperature rise; 0—do not perform calculation; 1—perform calculation.

7) `BulkModulus`: P—Isothermal bulk modulus at zero pressure. Must be consistent with the equation of state ($K_0 = \rho C_0^2$).

8) `ShearModulus`: P—Shear modulus (G_0 in Eq. 2).

9) `rho`: ML^{-3}—Reference material density (ρ_0).

10) `FlowStress`: P—Flow stress for J_2 plasticity (τ_0 in Eq. 12).

11) `HardeningModulus`: P—Isotropic linear hardening modulus for J_2 plasticity (H in Eq. 12).

12) `InitialPlasticStrain`: Unitless—Initial plastic strain for J_2 plasticity calculation. *This is a repetitive input and its meaning could change in future versions.*

13) `J2RelaxationTime`: τ—Timescale for visco-plastic relaxation of J_2 plasticity.

14) `NumCrackFamilies`: Integer—Number of flaw bins or families used to discretize the flaw size distribution (N_{bins} in Eq. 10).

15) `MeanFlawSize`: L—Mean flaw size used only for flaw-distribution types: 0 (delta) and 1 (Gaussian).

16) `FlawDensity`: L^{-3}—Average number of flaws per unit volume.

17) `StdFlawSize`: L—Standard deviation of the flaw size distribution. Used only in distribution type 1 (Gaussian).

18) `FlawDistType`: Integer—Distribution type for the probability distribution of flaw sizes; 0—Delta; 1—Gaussian; 2—Pareto. The majority of the model development has been done using a Pareto distribution.

19) `MinFlawSize`: L—Minimum flaw size for the flaw size distribution.

20) `MaxFlawSize`: L—Maximum flaw size for the flaw size distribution.

21) `FlawDistExponent`: Unitless—Exponent for the Pareto flaw size distribution.

22) `RandomizeFlawDist`: Unitless—Flag to apply flaw size distribution randomization during the first step of the simulation; 1—apply the randomization; 0—do not apply the randomization.

23) `RandomSeed`: Integer—One part of the random seed used to initialize the random number generator for setting the initial flaw size distribution.

24) `RandomizeMethod`: Integer—Set the method used to randomize the flaw size distribution. Choices are 1–7; 7 is the recommended choice, while 3 will cause each successively finer flaw bin to have twice the flaw density until all of the flaw bins are filled. When using Method 3, the smallest flaw may be smaller than `MinFlawSize` depending on the size of the element volume. Method 3 may be an approach to use fewer bins to cover the flaw distribution. See (Tonge 2014, app. B) for a complete description of the different randomization methods.

25) `BinBias`: Unitless—Scale the discretization of the flaw distribution to better capture small flaws. Only used when `RandomizeMethod` = 6 or 7.

26) `KIc`: $PL^{0.5}$—Fracture toughness for the micromechanics-based damage calculation (K_{IC} in Eq. 11).

27) `FlawFriction`: Unitless—Coefficient of friction for clean self contact. Used in the micromechanics calculation.

28) `FlawAngle`: Radians—Angle between the maximum compressive stress and the flaw normal. See (Tonge 2014, fig. 4.3) for an illustration.

29) `FlawGrowthExponent`: Unitless—Exponent for the micromechanics crack growth rate (γ_c in Eq. 11).

30) `FlawGrowthAlpha`: Unitless—Denominator for reference crack growth rate in micromechanics calculation (α_c in Eq. 11).

31) `CriticalDamage`: Unitless—Damage level to activate granular plasticity.

32) `MaxDamage`: Unitless—Maximum allowable damage level. Once damage reaches this value, the damage calculation is no longer performed.

33) `MicroMechPlaneStrain`: Unitless—Flag to select plane strain (1) or plane stress (0) for the micromechanics based damage calculation (recommend 1).

34) `MaxDamageInc`: Unitless—Maximum increment in damage if damage based sublooping is enabled.

35) `UseDamageTimestep`: Unitless—flag to allow the damage growth rate to determine the global time-step. *This is a legacy parameter that is no longer used. It should be set to 0.*

36) `dt_increaseFactor`: Unitless—Increase factor for the time-step when using the damage growth rate to compute the time-step size. *This is a legacy parameter that is no longer used. It should be set to 100.*

37) `IncInitDamage`: Unitless—Flag to include initial flaw size in the damage calculation; 0—do not include the initial flaw size in the calculation of the material point damage; 1—compute damage from both the initial flaw size and the wing crack size.

38) `DoFlawInteraction`: Unitless—Flag to activate the self consistent interaction calculation. This calculation can be a significant portion of the damage growth calculation time. This may be a possible place to trade accuracy for speed; 0—skip the calculation and use the element stress for the crack growth calculation; 1—perform the self consistent calculation.

39) `GPTimeConst`: τ—Relaxation time for granular flow.

40) `JLoc`: Unitless—Inelastic volume change ratio to activate the localized flag. Must be greater than 1.

41) `GPGranularSlope`: Unitless if `GPYieldSurfType` = 1, P if `GPYieldSurfType` = 2—Slope of the granular flow surface. Parameter A in Eq. 13 and Eq. 14. This input has units of P when used with `GPYieldSurfType`=2.

42) `GPCohesion`: P—Intersection of the granular flow surface with the hydrostatic stress axis (parameter B in Eq. 13 and Eq. 14).

43) `GPYieldSurfType`: Unitless—Switch between a cone (1) and parabola (2) type yield surface using the 2 surface representation. *The parabola (2) is not recommended.*

44) `GPPc`: P—Full consolidation pressure when using the 2 surface granular flow/pore compaction model.

45) `GPJref`: Unitless—Reference volume change ratio for the 2 surface granular flow/pore compaction model.

46) `GPPref`: P—Reference pressure for the 2 surface granular flow/pore compaction model.

47) `GPSurfaceType`: Unitless—Flag that controls the use of the single surface granular flow model (1) or the 2 surface (any value other than 1) model used in the original material model development. *The single surface model is under active development and may change in future releases. It is not recommended for production runs.*

48) `AbsToll`: Unitless—Absolute tolerance for single surface granular flow return calculation (recommend 1×10^{-14}).

49) `RelToll`: Unitless—Relative tolerance for single surface granular flow return calculation (recommend 1×10^{-8}).

50) `MaxIter`: Unitless—Maximum iterations for single surface granular flow return calculation (recommend 20).

51) `MaxLevels`: Unitless—Maximum number of recursive refinement levels for granular flow calculation. (This functionality can dramatically increase the model cost and does not significantly improve the granular flow update

accuracy. We recommend that this parameter be set to 1 to disable automatic refinement.)

52) `GFMSm0`: Unitless—Parameter μ_0 in Eq. 22.

53) `GFMSm1`: Unitless—Parameter μ_1 in Eq. 22.

54) `GFMSm2`: Unitless—Parameter μ_2 in Eq. 22.

55) `GFMSp0`: Unitless (less than 0)—Parameter p_0 in Eq. 23. Saturation compaction mean stress at large values of ϵ_p^v. This is the mean stress required to begin compressing out porosity for the fully distended material. This value should be almost 0.

56) `GFMSp1`: P—Parameter p_1 in Eq. 23.

57) `GFMSp2`: Unitless—Parameter p_2 in Eq. 23.

58) `GFMSp3`: Unitless—Parameter p_3 in Eq. 23.

59) `GFMSp4`: Unitless ($0 < p_4 < 1$)—Parameter p_4 in Eq. 24.

60) `GFMSa1`: P—Parameter a_1 in Eq. 20. Hydrostatic intercept for linear portion of F_f.

61) `GFMSa2`: P^{-1}—Parameter a_2 in Eq. 20. Nonlinear portion of F_f inside the exponent.

62) `GFMSa3`: P—Parameter a_3 in Eq. 20. Exponential scale for F_f.

63) `GFMSBeta`: Unitless ($0 < \beta \leq 1.0$)—Scale value to introduce nonassociative behavior into the bulking response of the material.

64) `GFMSPsi`: Unitless—Ratio of the triaxial tensile strength to the triaxial compressive strength (Brannon et al. 2009).

65) `GFMSJ3Type`: Unitless—Select the form for the Lode angle dependence (Brannon et al. 2009); 0—Drucker-Prager; 1—Gudehus; 2—William-Warnke.

66) `ArtVisc1`: Unitless—Linear artificial bulk viscosity coefficient.

67) `ArtVisc2`: Unitless—Quadratic artificial bulk viscosity coefficient.

68) `MGC0`: L/τ—Bulk sound speed at reference conditions for Mie–Grüneisen equation of state.

69) `MGGamma0`: Unitless—Grüneisen coefficient for equation of state.

70) `MGS1`: Unitless—Slope of $U_s - U_p$ curve.

71) `MGS2`: Unitless—Parameter for equation of state.

72) `MGS3`: Unitless—Parameter for equation of state.

73) `MGCv`: $EM^{-1}K^{-1}$—Specific heat per unit mass.

74) `MGTheta`: T—Reference temperature.

75) `JMin`: Unitless—Unused.

76) `MGNPts`: Unitless—Unused.

Not all constants are used in all computations. Constants 10–13 are only used if `usePlasticity` = 1. Constants 15–38 are only used if `useDamage` = 1. Within these constants, 15 and 17 are only used if `FlawDistType` = 0 or 1. Constants 21 and 25 only apply when `FlawDistType` = 2. Constants 39–65 only apply when `useGranularPlasticity` = 1. Constants 41–46 apply when `GPSurfaceType` = 0. Constants 48–65 apply when `GPSurfaceType` = 1.

3. Material Model State Variables

This material model computes the evolution of the total Cauchy stress (the deviatoric and isotropic parts) following the Abaqus UMAT conventions. In addition to updating the Cauchy stress, the model tracks the following internal-state variables. They are stored in an array, but are given names if the host code allows (this is not a standard capability in an Abaqus UMAT).

1) `Iel` ($\bar{I}_{el} = \frac{1}{3}\text{tr}(\bar{\boldsymbol{b}}_{el})$) : Unitless—Must be greater or equal to 1.0.

2) `damage` (D): Unitless—Must be positive and is no greater than `MaxDamage`.

3) `JGP` ($J_{GP} \approx \exp\left(\int_0^t \text{tr}(\boldsymbol{d}_{GP})d\tau\right)$): Unitless—Inelastic volume change ratio associated with granular flow.

4) `GP_strain` ($\gamma_{GP} = \int_0^t ||\boldsymbol{d}_{GP}^{\text{dev}}||d\tau$): Unitless—Accumulated deviatoric inelastic granular flow.

5) `GP_energy` ($\int_0^t J\boldsymbol{\sigma} : \boldsymbol{d}_{GP}d\tau$): EL^{-3}—Energy dissipated by granular flow per unit reference volume.

6) `plasticStrain` ($\gamma_p = \int_0^t ||\boldsymbol{d}_p||d\tau$): Unitless—Equivalent plastic strain for the J_2 type plastic flow.

7) `plasticEnergy` ($\int_0^t \dot{\gamma}_p||J\boldsymbol{\sigma}^{\text{dev}}||d\tau$): EL^{-3}—Energy dissipated through J_2 plasticity per unit reference volume.

8) `artViscPres`: P—Artificial viscous pressure that should be added to the stress tensor.

9) `volHeating`: $T\tau^{-1}$—Rate of change of the temperature.

10) `localized`: Unitless—Flag that is set when localization conditions are met: 1—Damage is greater than or equal to `CriticalDamage`; 2—Granular flow localization conditions met (`JGP > JLoc`); 3—Conditions for 1 and 2 were both met; 4—Localization caused by J_2 flow (the computed strength due to hardening is 0); 5—Conditions 1 and 4 were met; 6—Conditions 2 and 4 were met; 7—All 3 localization conditions were met (1, 2, and 4).

11) `epsVGP` ($\epsilon_{GP}^v = \int_0^t \text{tr}(\boldsymbol{d}_{GP})d\tau$): Unitless—Inelastic volume strain. Only used when `GPYieldSurfType`=1.

12) `gamGP` ($\gamma_{GP} = \int_0^t ||\boldsymbol{d}_{GP}^{\text{dev}}||d\tau$): Unitless—Accumulated Inelastic deviatoric strain. Only used when `GPYieldSurfType`=1.

13) `epsVGP_qs`: Unitless—Same as `epsVGP`, but for the rate-independent solution. This is only used when the single surface model (`GPYieldSurfType`=1) is used with rate-dependent granular flow (`GPTimeConst`> 0). The viscoplasticity algorithm (Brannon 2007) tracks both the actual material state and the state that represents the rate-independent solution.

14) `gamGP_qs`: Unitless—Same as `gamGP`, but for the rate-independent solution. This is only used when the single surface model (`GPYieldSurfType`=1) is used with rate-dependent granular flow (`GPTimeConst`> 0). The viscoplasticity algorithm (Brannon 2007) tracks both the actual material state and the state that represents the rate-independent solution.

15) `sig_qs_11`: P—Stress associated with the rate-independent solution to the granular-flow evolution. This is only used when the single surface model (`GPYieldSurfType`=1) is used with rate-dependent granular flow (`GPTimeConst`> 0). The visco-plasticity algorithm (Brannon 2007) tracks both the actual material state and the state that represents the rate-independent solution.

16) `sig_qs_22`: P—Stress associated with the rate-independent solution to the granular-flow evolution. This is only used when the single surface model (`GPYieldSurfType`=1) is used with rate-dependent granular flow (`GPTimeConst`> 0). The visco-plasticity algorithm (Brannon 2007) tracks both the actual material state and the state that represents the rate-independent solution.

17) `sig_qs_33`: P—Stress associated with the rate-independent solution to the granular-flow evolution. This is only used when the single surface model (`GPYieldSurfType`=1) is used with rate-dependent granular flow (`GPTimeConst`> 0). The visco-plasticity algorithm (Brannon 2007) tracks both the actual material state and the state that represents the rate-independent solution.

18) `sig_qs_23`: P—Stress associated with the rate-independent solution to the granular-flow evolution. This is only used when the single surface model (`GPYieldSurfType`=1) is used with rate-dependent granular flow (`GPTimeConst`> 0). The visco-plasticity algorithm (Brannon 2007) tracks both the actual material state and the state that represents the rate-independent solution.

19) `sig_qs_13`: P—Stress associated with the rate-independent solution to the granular-flow evolution. This is only used when the single surface model (`GPYieldSurfType`=1) is used with rate-dependent granular flow (`GPTimeConst`> 0). The visco-plasticity algorithm (Brannon 2007) tracks both the actual material state and the state that represents the rate-independent solution.

20) `sig_qs_12`: P—Stress associated with the rate-independent solution to the granular-flow evolution. This is only used when the single surface model (`GPYieldSurfType`=1) is used with rate-dependent granular flow

(GPTimeConst> 0). The visco-plasticity algorithm (Brannon 2007) tracks both the actual material state and the state that represents the rate-independent solution.

21) flawNumber_k: L^{-3}—Flaw number density in bin number k (ω_k).

22) starterFlawSize_k: L—Initial flaw size in bin number k (s_k).

23) wingLength_k: L—Flaw growth in bin number k (l_k).

Unused state variables are set to 0. The material model also updates the temperature of the material in the TEMP function argument for the Abaqus UMAT interface.

4. Key Physical Equations

4.1 Elastic Response

The model assumes a decoupled representation of the Kirchhoff stress tensor:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{dev} - p_s J_e \boldsymbol{I}. \tag{1}$$

The deviatoric stress $\boldsymbol{\tau}_{dev}$ is a linear function of the deviatoric part of the volume preserving elastic deformation as measured by $\bar{\boldsymbol{b}}_e = J_e^{-2/3}\boldsymbol{F}_e\boldsymbol{F}_e^T$:

$$\boldsymbol{\tau}_{dev} = G\left(\bar{\boldsymbol{b}}_e - \frac{1}{3}\mathrm{tr}(\bar{\boldsymbol{b}}_e)\boldsymbol{I}\right). \tag{2}$$

Here G is the damaged shear modulus defined as

$$G(D) = \left(G_0^{-1} + \frac{2D}{15}\left(3Z_r + 2Z_n - 4Z_c\right)\right)^{-1}. \tag{3}$$

The parameters Z_n, Z_r, and Z_c are functions of the elastic moduli and relate to the compliance of an individual crack.

The volumetric response is determined by a Mie–Grüneisen equation of state:

$$p_s(J_e,\theta) = \frac{K(D)}{K_0}p_H(J_e)\left[1 - \frac{\Gamma_0}{2}(1-J_e)\right] + \rho_0\Gamma_0\left[e_c(J_e) + c_\eta(\theta-\theta_0)\right] \tag{4}$$

$$p_H(J_e) = \begin{cases} \frac{\rho_0 C_0^2(1-J_e)}{(1-S_1(1-J_e)-S_2(1-J_e)^2-S_3(1-J_e)^3)^2} & \text{if } J_e < 1 \\ \rho_0 C_0^2(1-J_e) & \text{otherwise} \end{cases}. \tag{5}$$

The scaling factor K/K_0 softens the bulk modulus in response to damage growth. The damaged bulk modulus is defined as

$$K(D) = \left(K_0^{-1} + D\left(Z_n + 4Z_c\right)\right)^{-1}. \tag{6}$$

Here K_0 is the undamaged bulk modulus ($-\partial p_s/\partial J_e|_{D=0}$). The cold energy e_c is the solution to

$$\frac{de_c}{dJ_e} + \frac{\Gamma}{J_e}(e_c - e_0) = \frac{p_H}{\rho_0}\left(1 - \frac{\Gamma}{2J_e}(1 - J_e)\right). \tag{7}$$

Here $e_0 = c_\eta \theta_0$ and the initial condition is $e_c(1) = 0$. At finite temperature ($\theta_0 > 0$) the solution for e_c is not strictly positive. Since the reported strain energy `SSE` is e_c added to the deviatoric component ($G(D)\left(\text{tr}(\bar{\boldsymbol{b}}_{\text{el}}) - 3\right)/2$), the reported strain energy may not be positive for all deformations.

The artificial viscosity formulation is the linear quadratic viscosity commonly used in many hydrocodes:

$$p_{\text{visc}} = \begin{cases} (A_1 C_0 |\text{tr}(\boldsymbol{d}_e)| dx + A_2 \text{tr}(\boldsymbol{d}_e)^2 dx^2) & \text{if tr}(\boldsymbol{d}_e) \leq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{8}$$

The heating rate associated with the work done by the artificial viscosity is

$$\dot{\theta}_{\text{visc}} = \frac{-J p_{\text{visc}} \text{tr}(\boldsymbol{d}_e)}{\rho_o c_v}. \tag{9}$$

4.2 Micromechanics of Damage (useDamage = 1)

The model uses a micromechanics-based damage model where damage is defined as

$$D = \sum_{k=1}^{N_{\text{bins}}} \omega_k \left(s_k + l_k\right)^3. \tag{10}$$

In the summation, the bin number k loops over the N_{bins} that are used to discretize the local flaw size distribution; ω_k is the number density of flaws per unit volume that are represented by the flaw family k; the initial flaw size is s_k, which has grown an additional length l_k due to the applied loading history.

When `DoFlawInteraction` = 1, the model uses a self-consistent Eshelby in-

clusion solution to compute the elevated stress in the neighborhood of a flaw due to the surrounding flaws. This solution involves the solution to a linear system of 8 equations and can be disabled (at the expense of not including the interactions) by setting `DoFlawInteraction` = 0. After computing the effective stress around a flaw ($\boldsymbol{\sigma}_{\text{eq}}$), the stress-intensity factor for crack growth is computed based on a wing-cracking mechanism (Tonge 2014, eqs. 4.57 and 4.58). It depends on the crack-face coefficient of friction (`FlawFriction`), applied load ($\boldsymbol{\sigma}_{\text{eq}}$), the flaw angle (`FlawAngle`), and the representative flaw size s_k. From that stress-intensity factor (K_I), we compute the crack-growth rate using

$$\dot{l} = \frac{C_r}{\alpha_c} \left(\frac{K_I - K_{IC}}{K_I - 0.5 K_{IC}} \right)^{\gamma_c} . \tag{11}$$

The increased crack length is used to update the damage parameter.

4.3 Traditional J_2 Plasticity (usePlasticity = 1)

The J_2 volume preserving plasticity implementation uses a multiplicitive decomposition of the deformation gradient into elastic and plastic portions. It is integrated in the spatial configuration using an objective integration algorithm (Simo and Hughes 2000, ch. 9). Activation of this component may not be justified for all ceramic systems. It is an elastic-plastic material model with linear strain hardening. The yield surface is defined by

$$f(\boldsymbol{\tau}) = ||\boldsymbol{\tau}_{\text{dev}}|| - \left(\frac{G(D)}{G_0} \right) (H \epsilon_p + \tau_0) . \tag{12}$$

The factor $\frac{G}{G_0}$ corrects for the effect of damage growth where the effective yield stress decreases with damage in the same way that the shear modulus decreases with damage (the elastic strain required to cause yield is independent of damage). The hardening modulus may be either positive or negative (softening). When the material softens to zero strength, the `localized` flag is set to 4. *The coupling between this plasticity module and damage is not heavily tested.*

4.4 Granular Flow (useGranularPlasticity = 1)

Two separate granular-flow models are implemented in this material model version. The original model was discussed in (Tonge 2014, ch. 4) and uses a 2-surface approach to describe the distortion and compaction mechanisms for granular-flow. A

newer single-surface model has been added and is based on the tear-drop-shaped single-surface model (Fossum and Brannon 2006; Brannon et al. 2009).

4.4.1 Two-Surface Flow Compaction Model (GPSurfaceType = 0)

This 2-surface model computes the granular-flow and porosity increase first followed by the porosity compaction calculation. The compaction calculation does not depend on the deviatoric stress.

4.4.1.1 Granular Flow and Porosity Generation

Flow behavior is associative to the yield surface and there is no hardening and this flow formulation is written in terms of the Kirchhoff stress $\boldsymbol{\tau}$. There are 2 choices for the shape of the granular-flow-yield surface determined by the input `GPYieldSurfaceType`. Surface 1 is defined by

$$f(\boldsymbol{\tau}) = \sqrt{\boldsymbol{\tau}_{\text{dev}} : \boldsymbol{\tau}_{\text{dev}}} + A\left(\frac{\text{tr}(\boldsymbol{\tau})}{\sqrt{3}} - B\right). \tag{13}$$

Yield Surface 2 changes the units of A to pressure units and is defined by

$$f(\boldsymbol{\tau}) = \boldsymbol{\tau}_{\text{dev}} : \boldsymbol{\tau}_{\text{dev}} + A\left(\frac{\text{tr}(\boldsymbol{\tau})}{\sqrt{3}} - B\right). \tag{14}$$

A linear viscosity model is activated by setting `GPTimeConst` to a positive value. The linear viscosity model follows a Deviant–Louis-type viscoplasticity model that is unconditionally stable and only tracks the current state of the material (Simo and Hughes 2000, sec. 2.7.4).

The plastic flow direction is $\boldsymbol{m} = (\boldsymbol{n} + \zeta A\hat{\boldsymbol{I}})/\sqrt{1 + \zeta^2 A^2}$. Here $\boldsymbol{n} = \boldsymbol{\tau}_{\text{dev}}/||\boldsymbol{\tau}_{\text{dev}}||$, $\hat{\boldsymbol{I}} = \boldsymbol{I}/\sqrt{3}$, A is the tangent to the yield surface in $r - z$ space, and ζ is a parameter that is iteratively decreased from 1 to ensure positive plastic dissipation.

4.4.1.2 Pore Compaction

This module is turned on when granular plasticity is activated using the 2-surface model (`GPSurfaceType`=0). This is an additional yield surface that depends on only the hydrostatic pressure ($p = -1/3\text{tr}(\boldsymbol{\sigma})$).

It is defined by

$$f_\phi(P, J^{GP}, J) = \begin{cases} \frac{P}{P_c - P_0} - \frac{P_0}{P_c - P_0} \exp\left(-\frac{P_c - P_0}{2P_0(J_0^{GP}-1)}(J^{GP} - J_0^{GP})\right) & P < P_0 \\ (J^{GP} - 1) - (J_0^{GP} - 1)J^2 \left(\frac{P_c - P}{P_c - P_0}\right)^2 & P_0 \leq P < P_c \\ J^{GP} - 1 & P < P_c. \end{cases} \tag{15}$$

This is a simple porosity model with a quadratic crush behavior for pressures beyond P_0 and an exponential compaction behavior for lower pressures. The plastic flow resulting from pore compaction is purely isotropic.

4.4.2 Single Surface Flow Compaction Model (GPSurfaceType = 1)

This granular-flow model is based on the Kayenta yield and flow surface (Brannon et al. 2009). The yield surface is written in terms of the Lode invariants. For a Cauchy stress tensor $\boldsymbol{\sigma}$ with a deviatoric part $\boldsymbol{s} = \boldsymbol{\sigma} + p\boldsymbol{I}$, the Lode invariants are

$$r = \sqrt{\boldsymbol{s} : \boldsymbol{s}} \tag{16}$$

$$z = \frac{1}{\sqrt{3}}\text{tr}(\boldsymbol{\sigma}) \tag{17}$$

$$\sin 3\theta = \frac{J_3}{2}\left(\frac{3}{J_2}\right)^{2/3}. \tag{18}$$

The yield function (Fossum and Brannon 2006) is written as

$$f(r, z, \theta, \gamma_p, \epsilon_p^v) = \Gamma(\theta)^2 \frac{1}{2} r^2 - F_f(z, \gamma_p) \left|F_f(z, \gamma_p)\right| F_c(z, \epsilon_p^v) \tag{19}$$

$$F_f(z, \gamma_p) = \left(\frac{\mu(\gamma_p)}{\mu_0}\right)\left(a_1 - a_3 \exp\left(a_2\sqrt{3}z\right) - \mu_0\sqrt{3}z\right) \tag{20}$$

$$F_c(z, \epsilon_p^v) = \begin{cases} 1 - \frac{\left(\kappa(\epsilon_p^v) - \sqrt{3}z\right)^2}{\left(\kappa(\epsilon_p^v) - X(\epsilon_p^v)\right)^2} & \text{if } \sqrt{3}z < \kappa \\ 1 & \text{otherwise} \end{cases} \tag{21}$$

$$\mu(\gamma_p) = \mu_1 + (\mu_0 - \mu_1)\exp\left(-\mu_2\gamma_p\right) \tag{22}$$

$$X(\epsilon_p^v) = p_1\left(p_0 + \frac{1}{2}\left(\ln(p_3 + p_2\epsilon_p^v) - \left|\ln(p_3 + p_2\epsilon_p^v)\right|\right)\right) \tag{23}$$

$$\kappa(\epsilon_p^v) = p_4 X(\epsilon_p^v). \tag{24}$$

The plastic flow direction is

$$\boldsymbol{M} = \alpha(\boldsymbol{N}_{\text{dev}} + \beta\boldsymbol{N}_{\text{iso}}). \tag{25}$$

Here $\boldsymbol{N}_{\text{dev}}$ and $\boldsymbol{N}_{\text{iso}}$ are the deviatoric and isotropic parts of yield surface normal, β is an input parameter ($\beta \leq 1$), and α is a normalization parameter to ensure that $\boldsymbol{M} : \boldsymbol{M} = 1$ (Brannon and Leelavanichkul 2010, eq. 21). This granular-flow model uses the multistage return algorithm (Brannon and Leelavanichkul 2010).

5. Yield Surface Illustrations

For illustrative purposes, the micromechanics-based, damage-growth model can be viewed as providing an evolving damage-growth surface that depends on the extreme principal stresses. Such a surface is illustrated in Fig. 1. For a delta distribution of cracks, crack growth begins when the stresses exceed the blue line. Under compressive loading, the stress required to sustain damage growth increases until the stability limit (green line), then decreases with increasing damage until a user-specified maximum allowable damage level is reached. Figure 1 shows the stress history for a material point loaded at a constant strain rate under uniaxial stress conditions. The material point stress is plotted as a function of time. The peak in the blue curve represents the stability limit for this loading condition and the transition to the dotted green line represents the activation of granular flow when damage reaches the user-defined threshold of 0.125.

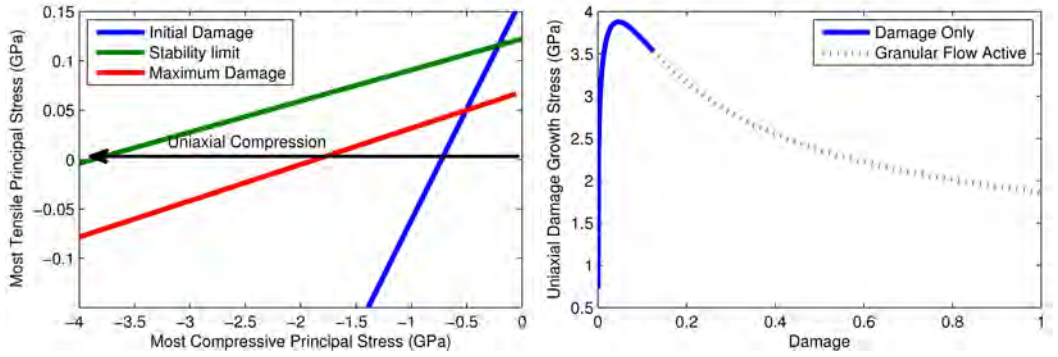

Fig. 1 a) shows the compressive stress states beyond which damage growth occurs, with the arrow indicating the path for uniaxial compression; b) plots uniaxial compressive stress required for damage growth as a function of damage level.

Once granular flow is activated, there is a competition between porosity growth and pore compaction. The pressure required to initiate pore compaction as a function of the distension (a measure of porosity) is shown in Fig. 2. The 2 different choices for `GPSurfaceType` use different representations of the crush curve, but either can be fit to experimental data if available.

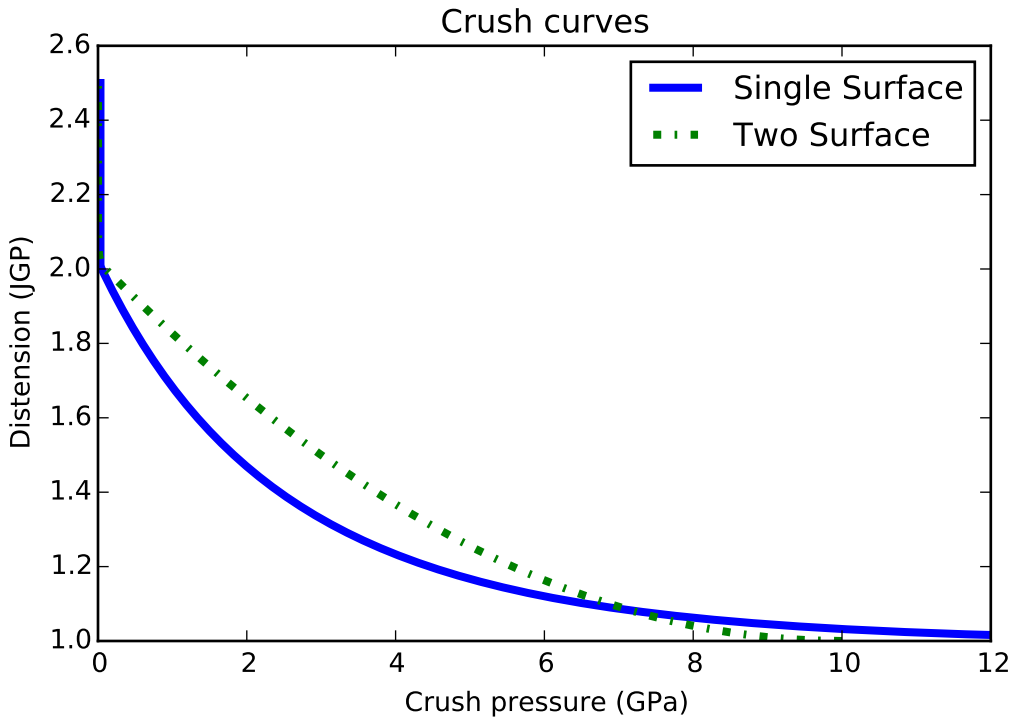

Fig. 2 Illustration of the crush curves showing pressure required to initiate pore collapse for pure hydrostatic loading as a function of distension for the 2 different granular-flow surface types

The single-surface representation for the granular-flow-yield surface is enabled by setting `GPSurfaceType=1`. This model provides a smooth representation of the increase in deviatoric strength with pressure until pore compaction begins to dominate, then provides a gradual reduction in strength with increasing pressure up to the crush pressure. The meridional profile of the yield surface is shown in Fig. 3. This model includes the option to specify Lode angle dependence, which reduces the strength of the material as the loading conditions deviate from triaxial compression. For different ratios of triaxial tensile strength to triaxial compressive strength, the octahedral profile is shown in Fig. 4.

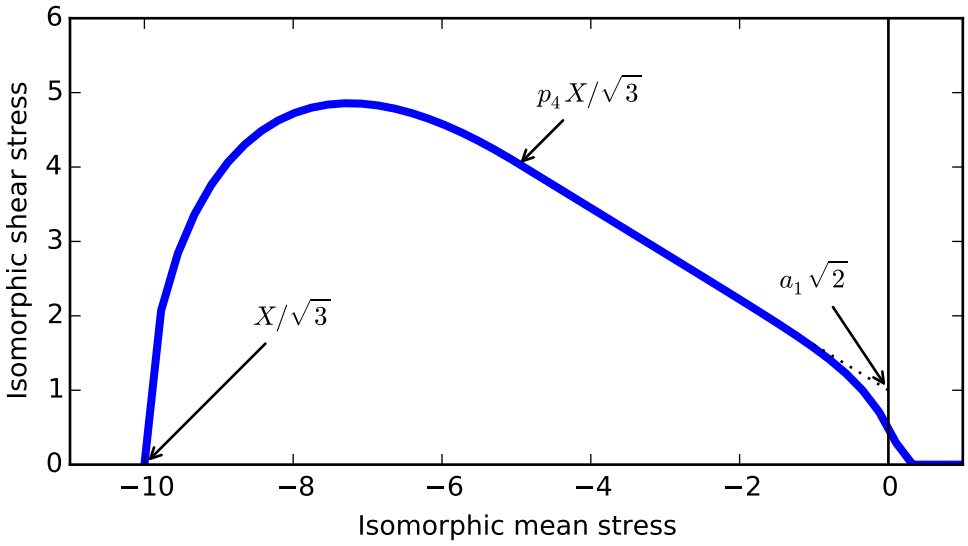

Fig. 3 Meridional profile of the yield surface for granular flow when using `GPSurfaceType=1`

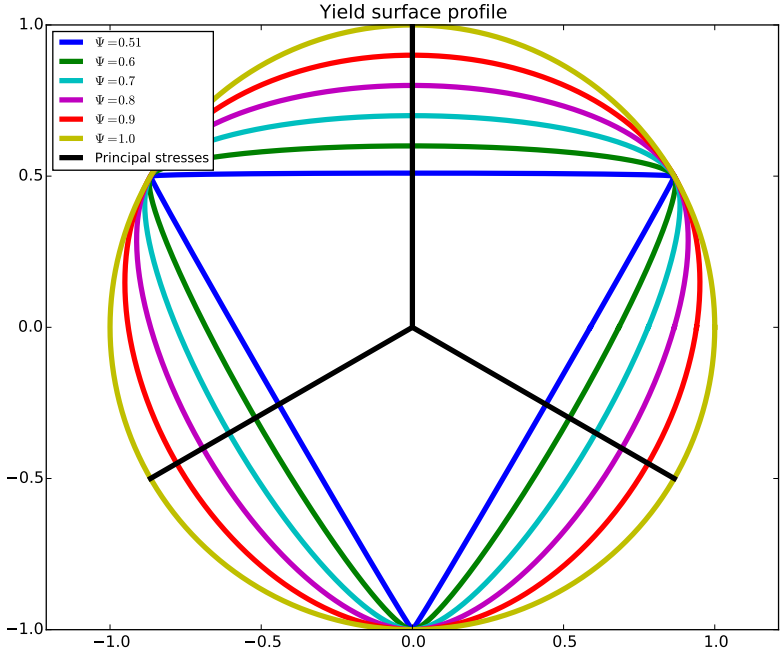

Fig. 4 Octahedral profile of the yield surface for granular flow when using `GPSurfaceType=1` **and** `GFMSJ3Type=2` **for different values of the** Ψ

6. Single Material Point Driver

The source package for the material model comes with a single element driver for the material model called `runUMATTR_MixedBC`. This small program drives the material model using a user-defined combination of stress- and strain-rate boundary conditions as a function of time. This driver is called using this command line:

```
./runUMATTR_MixedBC <Boundary File> <Parameter File> <History File>
```

The `Boundary File` is a text file containing white-space-separated numbers. The first line contains the time-step size and the representative edge length followed by 6 flags (0 or 1) where 0 denotes an rate of deformation (symmetric part of the velocity gradient) and 1 denotes an applied Cauchy stress-boundary condition for each of the symmetric tensor components in the order XX YY ZZ YZ XZ XY. The subsequent lines contain a time followed by the boundary conditions. The applied boundary conditions are treated as piecewise constant; they are not interpolated between time points in the table. The simulation ends at the last time in the table.

The `Parameter File` is a text file that contains all of the material parameters listed in Section 2 one parameter per line with the tag followed by white space followed by the value for that parameter. The `History File` is an output file with column headers on the first line then fixed-width columns of all history variables at every simulation time step in the subsequent lines.

7. Model Test Suite

The granular-flow algorithm is tested against 3 analytic solutions (Brannon and Leelavanichkul 2010). The first of 3 analytic solutions tests an associative Drucker–Prager material subjected to plastic loading that causes a rotation of the principal directions of the stress tensor while maintaining constant, principal stress values. The results from this verification test are shown in Fig. 5.

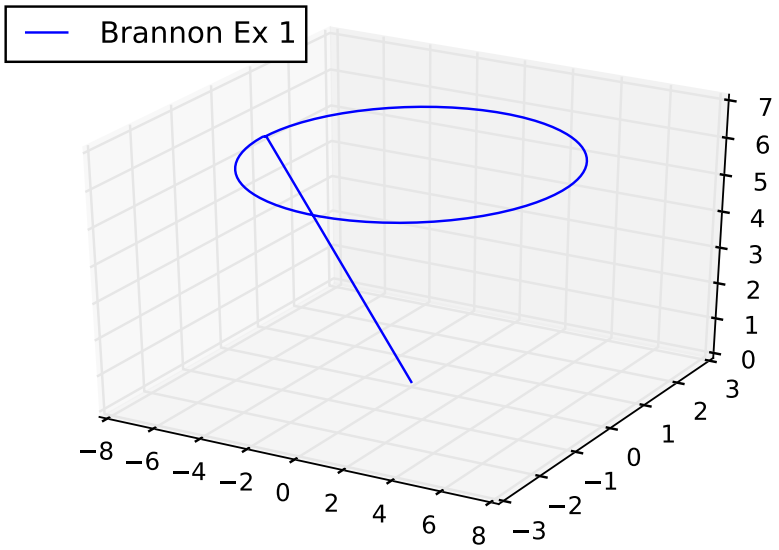

Fig. 5 Depiction of the loading path in a stationary 3-dimensional manifold for stress space

The second problem tests a J_2 Von–Mises material under plastic loading that is not coaxial to the stress deviator. A graphical comparison of the simulation results with the analytic solution is provided in Fig. 6.

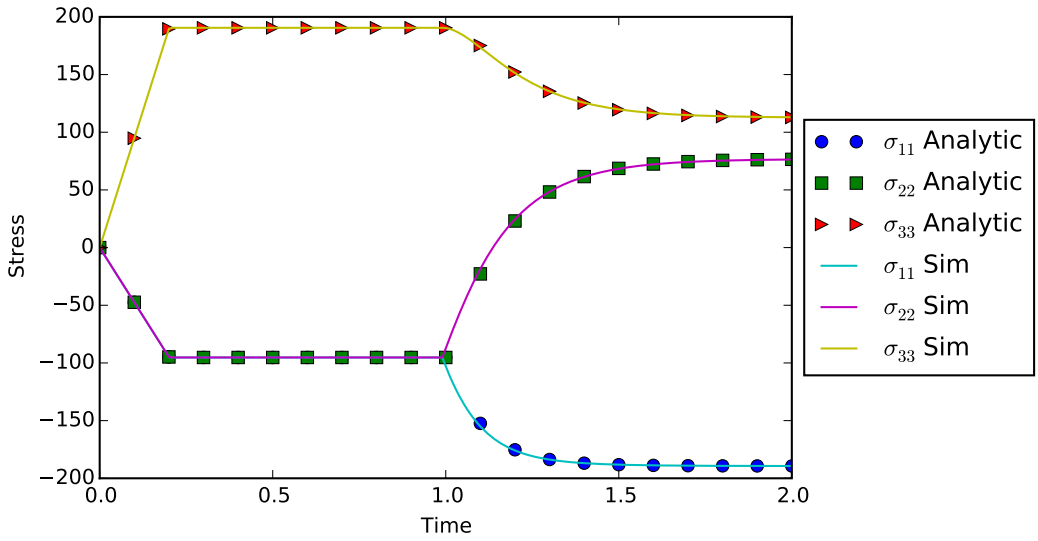

Fig. 6 Comparison between the simulated stress components and the analytic solution under nonproportional loading for a Von–Mises material

The third test problem is a 5-stage loading path for an associative Drucker–Prager material expanded from the published version (Brannon and Leelavanichkul 2010) to include a leg that traverses around the hydrostatic tensile vertex. The expanded solution for associative flow was provided by M. Hommel. The first stage is hydrostatic compression. The next leg of the problem is triaxial extension loading (the 2 equal principal values are greater than the third) such that continued loading after yield results in a stationary stress. The loading direction is then changed so that the stress crosses the yield surface and approaches the yield surface in the normal direction. Upon yield, the stress travels along the yield surface to higher deviatoric and hydrostatic stresses. The next leg is plastic loading to the hydrostatic tensile vertex. After reaching the hydrostatic tensile vertex, plastic loading takes the stress back to the location where yield occurred during the second leg of the test. The results from the simulation are shown in Figs. 7 and 8. The results do not exactly match the analytic solution; however, component tests that isolated the granular-flow portion of the model demonstrated significantly better agreement when the bulk modulus and shear modulus were exactly constant and the additive decomposition of the strain increment was consistent with the total strain measure. The use of a Mie–Grüneisen equation of state, the coupling between the bulk modulus and the distension, and the geometric nonlinearity introduced by the use of $\bar{\boldsymbol{b}}_{\text{el}}$ to define the deviatoric stress all

contribute to the differences between the analytic solution and the simulation results shown in Figs. 7 and 8.

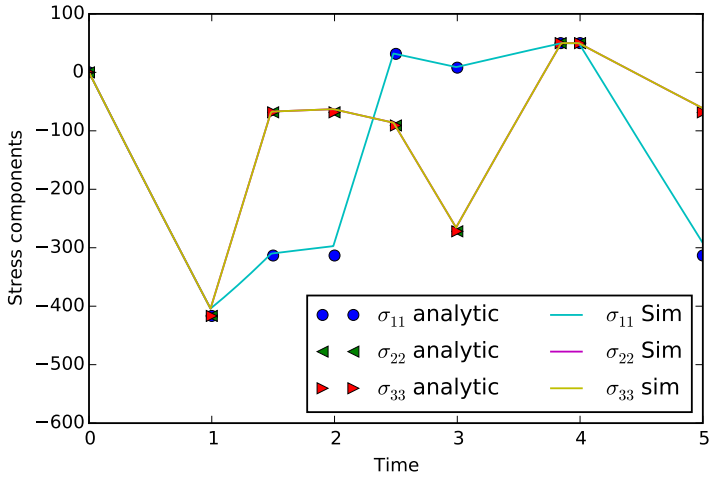

Fig. 7 Comparison between the simulated stress components and the analytic solution for the 5-leg loading path

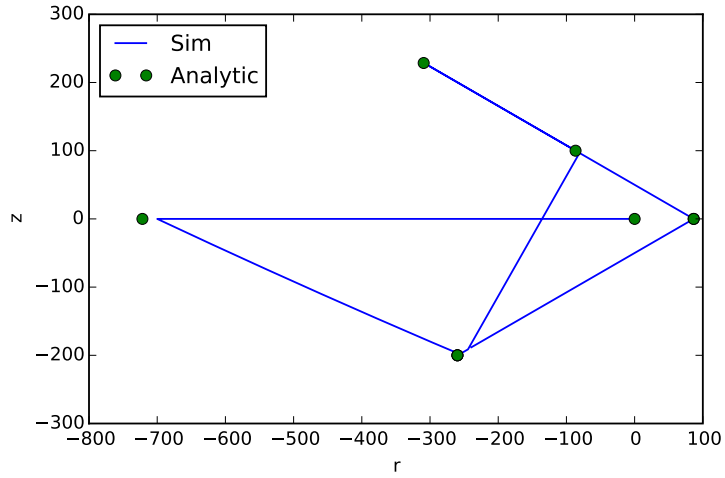

Fig. 8 Illustration of the simulated and input stress path in a signed deviatoric stress versus hydrostatic stress space

Verification of the material model's implementation in the host code was performed by testing simple shear at a shear rate of 10,000 1/s in all 6 directions (12, 13, 21, 23, 31, 23). The stress history from a representative simple shear simulation is shown

in Fig. 9. Shear in the other 6 directions produced similar results. All other history variables show similar agreement among the different host codes.

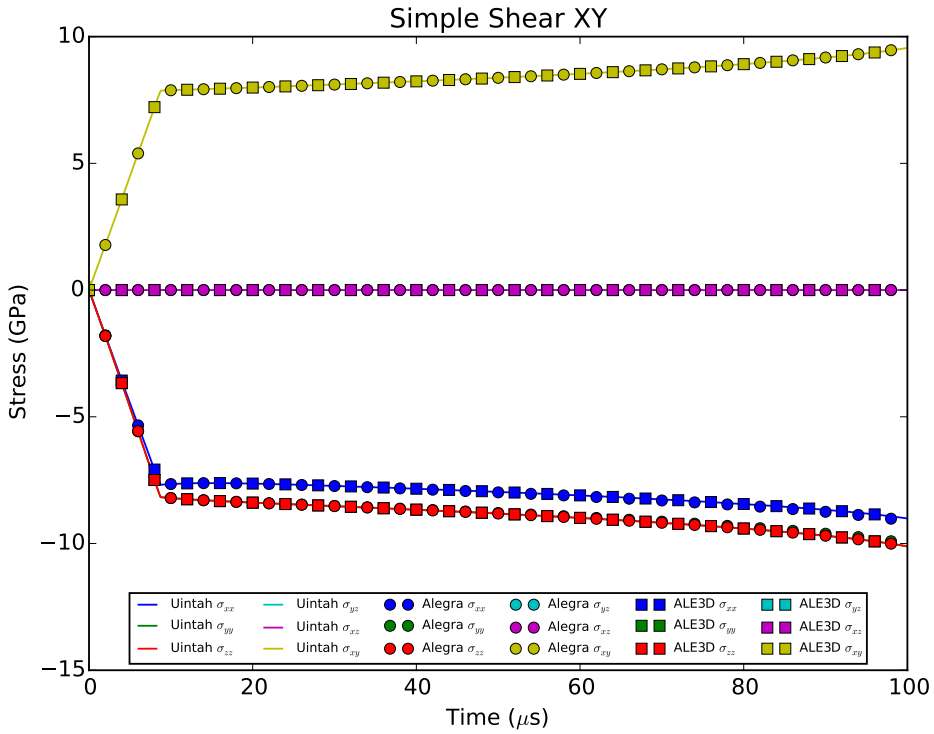

Fig. 9 Stress history for representative single-element simulations showing multiple host codes producing nearly identical results for simple shear to a shear of 1

8. Model Interface Notes

The model uses an Abaqus UMAT interface, which has a function prototype of

```
void PTR_umat_stressUpdate(
  double STRESS[6], double STATEV[], double DDSDDE[6][6],
  double *SSE, double *SPD, double *SCD, double *RPL,
  double DDSDDT[6], double DRPLDE[6], double *DRPLDT,
  const double STRAN[6], const double DSTRAN[6], const double TIME[2],
  const double *DTIME, double *TEMP, double *DTEMP,
  const double *PREDEF, const double *DPRED, const double *CMNAME,
  const int *NDI, const int *NSHR, const int *NTENS, const int *NSTATV,
  const double PROPS[], const int *NPROPS, const double COORDS[3],
  const double DROT[3][3], double *PNEWDT, const double *CELENT,
  const double DFGRD0[3][3], const double DFGRD1[3][3],
  const int *NOEL, const int *NPT, const int *LAYER,
  const int *KSPT, const int *KSTEP, const int *KINC);
```

The following exception types may be thrown by the constitutive model if some invalid conditions are encountered (like negative plastic work):

- `runtime_error`
- `out_of_range`—should not happen. Indicates an incorrect matrix index.
- `domain_error`—Matrix math error, either inversion of a singular matrix or division by zero.

Currently all implementations allow thrown exceptions to propagate and cause the host code to crash.

Total internal energy can be computed from `SSE`+`MGCv`* (`TEMP-MGTheta`).

`DROT` is the incremental rotation that is consistent with the stress-integration algorithm. `STRAN` is an approximation to the logarithmic strain, but in this model only the 3 normal terms are used to compute the volume-change ratio. To compute the total strain, one can integrate the strain rate using the same objective algorithm used to integrate the stress. Total plastic work per unit of mass is `SPD`. This is the same as (`plasticEnergy`+`GP_energy`)/`rho`. Axisymmetric geometry is not supported because it is inconsistent with the failure modes in brittle materials. The longitudinal sound speed is `sqrt(DDSDDE[0][0]/rho_cur)`.

The function `PTR_umat_stressUpdate_ALE3D` tracks and updates the temperature in the `SCD` variable place holder. This is a work-around for host codes that do not accurately track the temperature being fed into the UMAT.

References

Brannon RM, Fossum A, Strack O. KAYENTA: Theory and user's guide. Report No.: SAND2009-2282. Albuquerque (NM): Sandia National Laboratories (US). 2009 Mar.

Brannon RM. In: Horie Y, editor, ShockWave Science and Technology Reference Library: Volume 2. Solids I. Berlin (Germany): Springer; 2007. Chapter 6; p. 189–274.

Brannon RM, Leelavanichkul S. A multi-stage return algorithm for solving the classical damage component of constitutive models for rocks, ceramics, and other rock-like media. Int J Frac. 2010;163(1–2):133–149. doi:10.1007/s10704-009-9398-4.

Fossum AF, Brannon R. On a viscoplastic model for rocks with mechanism-dependent characteristic times. Acta Geotechnica. 2006;1(2):89–106. doi: 10.1007/s11440-006-0010-z.

Simo JC, Hughes T. Computational inelasticity. Volume 7. Interdisciplinary applied mathematics. 2nd ed. Berlin (Germany): Springer; 2000.

Tonge AL. A unified framework which uses multi-scale microstructure information for modeling dynamic failure in brittle materials [dissertation].[Baltimore (MD)]: The Johns Hopkins University; 2014.

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
3 (PDF)	JOHNS HOPKINS UNIVERSITY N DAPHALAPURKAR K RAMESH L GRAHAM-BRADY
1 (PDF)	RUTGERS R HABER
1 (PDF)	UNIV CALIFORNIA SANTA BARBARA R MCMEEKING

ABERDEEN PROVING GROUND

32 (PDF)
DIR USARL
RDRL WM
B FORCH
J MCCAULEY
RDRL WML H
B SCHUSTER
RDRL WMM
J BEATTY
RDRL WMM B
G GAZONAS
D HOPKINS
B LOVE
B POWERS
RDRL WMM E
J SWAB
RDRL WMM F
T SANO
M TSCHOPP
RDRL WMM G
J ANDZELM
RDRL WMP
S SCHOENFELD
RDRL WMP A
S BILYK
RDRL WMP B
C HOPPEL
S SATAPATHY
M SCHEIDLER
A SOKOLOW
T WEERASOORIYA
RDRL WMP C
R BECKER
T BJERKE
D CASEM
J CLAYTON
M GREENFIELD
R LEAVY
J LLOYD
S SEGLETES
A TONGE
C WILLIAMS
RDRL WMP D
R DONEY
C RANDOW
RDRL VTM
M HAILE